

7

## ENCODING METHOD FOR DECODING SIGNAL

Publication number: JP10112865

Publication date: 1998-04-28

Inventor: SHIMIZU ATSUSHI; SAGATA ATSUSHI; WATANABE YUTAKA

Applicant: NIPPON TELEGRAPH &amp; TELEPHONE

Classification:

- International: H04N7/32; H03M7/36; H04N7/32; H03M7/36; (IPC1-7): H04N7/32; H03M7/36

- European:

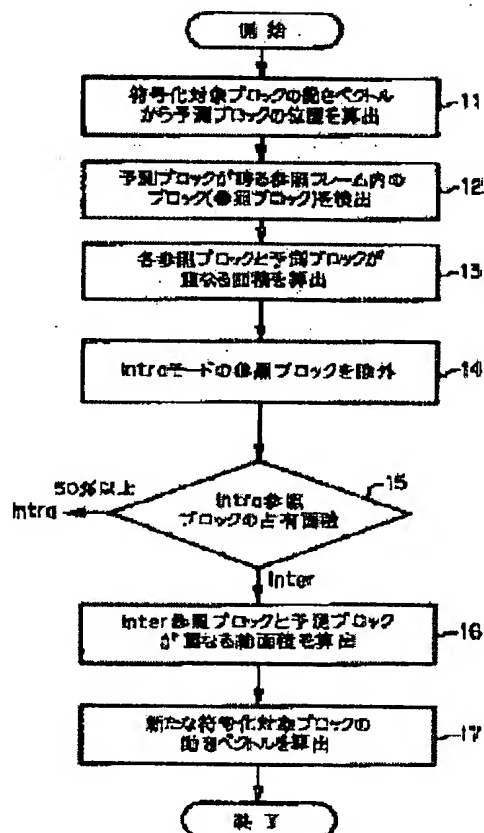
Application number: JP19960264122 19961004

Priority number(s): JP19960264122 19961004

Report a data error here

## Abstract of JP10112865

**PROBLEM TO BE SOLVED:** To thin a reference frame without performing motion detection in recoding and also to scale encoded data that suppresses image degradation without increasing an arithmetic amount as a result. **SOLUTION:** The position of a predictive block is calculated from a motion vector of an encoding object block. More than one reference blocks in reference frames that are spanned by the predictive block are detected. An area where the predictive block and each reference block are superimposed on each other is calculated. The encoding mode of each reference block is examined and the blocks of Intra-mode are excluded. The total area that is occupied by the reference block of the Inter-mode is calculated. A motion vector of a new encoding object block is calculated from the occupied area of each Inter-mode reference block, the motion vector of each reference block and the motion vector of the encoding object block.



Data supplied from the esp@cenet database - Worldwide

(51) Int.Cl.<sup>6</sup>

識別記号

F I

H 0 4 N 7/32

H 0 4 N 7/137

Z

H 0 3 M 7/36

H 0 3 M 7/36

審査請求 未請求 請求項の数 2 O L (全 7 頁)

(21) 出願番号

特願平8-264122

(22) 出願日

平成8年(1996)10月4日

(71) 出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72) 発明者 清水 淳

東京都新宿区西新宿三丁目19番2号 日本  
電信電話株式会社内

(72) 発明者 嵯峨田 淳

東京都新宿区西新宿三丁目19番2号 日本  
電信電話株式会社内

(72) 発明者 渡辺 裕

東京都新宿区西新宿三丁目19番2号 日本  
電信電話株式会社内

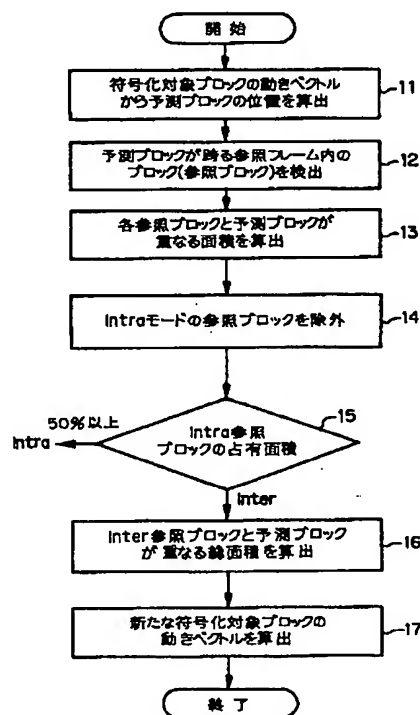
(74) 代理人 弁理士 若林 忠

(54) 【発明の名称】 復号信号の符号化方法

(57) 【要約】

【課題】 再符号化時に、動き検出をすることなく参照フレームを間引くことができ、結果として演算量を増加させず、かつ、画質劣化を抑えた符号化データのスケールリングを可能にする。

【解決手段】 符号化対象ブロックの動きベクトルから予測ブロックの位置を算出する。予測ブロックがまたがる参照フレーム内の1つ以上の参照ブロックを検出する。予測ブロックと各参照ブロック  $block_i$  が重なりあう面積  $pi$  を算出する。各参照ブロック  $block_i$  の符号化モードを調べ、Intraモードのブロックを除外する。Interモードの参照ブロック  $Inter-block_i$  が占める総面積  $P_{all}$  を計算する。各Interモード参照ブロックの占有面積  $Inter-pi$ 、各参照ブロックの動きベクトル、符号化対象ブロックの動きベクトルから新たな符号化対象ブロックの動きベクトルを算出する。



## 【特許請求の範囲】

【請求項1】 あるビットレートで符号化された符号化データの復号信号を符号化する方法において、元の符号化データに含まれるデータから再符号化対象ブロックの予測ブロックがまたがる動き補償参照フレームのブロックを検出する段階と、該動き補償参照フレームのブロックが持つ動きベクトルを基に新たな動きベクトルを算出する段階とを有することを特徴とする、復号信号の符号化方法。

【請求項2】 再符号化対象フレームの予測ブロックがまたがる動き補償参照フレームのブロックが持つ動きベクトルと、予測ブロックと動き補償参照フレーム内の各ブロックとが重なる面積を基に重み係数を算出し、新たな動きベクトルを算出する請求項1記載の復号信号の符号化方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、あるビットレートで符号化された符号化データの復号信号を符号化する方法に関する。

## 【0002】

【従来の技術】あるビットレートにて符号化された符号化データから、複数のビットレートの符号化データを生成する一方法として、再生画像を再符号化する方法がある。図7にシステム図を示す。

【0003】図7において、再生画像だけを用いて再符号化する場合は、通常の符号化処理と同等である。通常、符号化処理において、動きベクトルを算出する動き探索処理に最も演算量を必要とする。そこで、動きベクトルを符号化データから抽出して利用することで、再符号化時の演算量を削減できる。さらに、符号化モードについても、元の符号化データと同じモードを用いることで、演算量を削減できる。逆に、符号化モードを再符号化時に新たに決定すれば、符号化効率の低下を抑えることができる。

【0004】符号化データから抽出した符号化パラメータを用い、新たに与えられたビットレートにて符号化処理を行なうことにより、通常の符号化処理に比べ、少ない演算量で符号化データ量の縮小が可能となる。

【0005】また、再符号化方式では、符号化データの一部を削除する形式と異なり、符号化側と復号側で異なる参照画像を用いることに起因するドリフトは発生しない。

## 【0006】

【発明が解決しようとする課題】従来方法では、再生画像を再符号化する時に量子化ステップを大きくし、粗く量子化することで、発生情報量を小さくしている。符号化データは、大きく符号化モードや動きベクトルなどのオーバーヘッド情報と直交変換係数情報に分けられる。従来の方法で発生情報量を小さくできるのは、直交変換

係数情報でしかない。このため、元のビットストリームのオーバーヘッドが大きい場合やスケーリング（縮小）する目標ビットレートが極端に小さい場合は、スケーリングができない可能性がある。仮に、可能であったとしても、直交変換係数情報が殆んど失われてしまうため、画質劣化が顕著になる。

【0007】そこで、再符号化する際に、全ての再生画像を再符号化せず、一部の再生画像だけを再符号化することにより、上記の問題は回避できる。

【0008】しかし、演算量削減のため、動きベクトルを符号化データから抽出している場合、その動きベクトルを算出した際の参照フレームを間引くことはできない。動きベクトルは、符号化対象フレームと参照フレームから算出するが、仮に、その参照フレームが存在しない場合、正しい動き補償ができなくなり、符号化効率が極端に低下する。

【0009】このように、量子化ステップによる再符号化では不十分な場合、従来の方法では、画質劣化が顕著になり、最悪スケーリングが不可能になる場合があった。

【0010】本発明の目的は、再符号化時に、動き検出をすることなく参照フレームを間引くことができ、結果として演算量を増加させず、かつ、画質劣化を抑えた符号化データのスケールが可能な、復号信号の符号化方法を提供することにある。

## 【0011】

【課題を解決するための手段】本発明の、復号信号の符号化方法は、元の符号化データに含まれるデータから再符号化対象ブロックの予測ブロックがまたがる動き補償参照フレームのブロックを検出する段階と、該動き補償参照フレームのブロックが持つ動きベクトルを基に新たな動きベクトルを算出する段階とを有する。

【0012】本発明の実施態様によれば、再符号化対象フレームの予測ブロックがまたがる動き補償参照フレームのブロックが持つ動きベクトルと、予測ブロックと動き補償参照フレーム内の各ブロックとが重なる面積を基に重み係数を算出し、新たな動きベクトルを算出する。

【0013】本発明では、動き補償の参照フレームを間引き、符号化データ内の動きベクトルから新たに動きベクトルを算出する。

【0014】図5に示す参照フレームを間引く場合を考える。図5では、フレームn+1はフレームn+2の参照フレームとなっている。よって、フレームn+1を間引く場合、フレームn+2は、フレームnを参照フレームとするように変更する必要がある。

【0015】図6に、符号化対象ブロックと参照フレームの関係を示す。以下に符号化対象ブロックの動きベクトルの算出方法について述べる。

【0016】まず、再符号化の符号化対象ブロックの動きベクトル

3

【0017】

【外1】

 $\bar{v}_0$ 

から、間引く参照フレーム内での予測ブロックの位置を求める。予測ブロックの位置に存在する1つ以上のブロック（参照ブロック） $block_i$  ( $i=1, 2, \dots$ ) が持つ動きベクトル

【0018】

【外2】

 $\bar{v}_i$ 

を符号化データから抽出する。この動きベクトルは、予測ブロックと間引く参照ブロックの配置により複数存在する。図6の例では、予測ブロックが4つの参照ブロック $block_1, block_2, block_3, block_4$ にまたがっており、4つの動きベクトル

【0019】

【外3】

 $\bar{v}_1, \bar{v}_2, \bar{v}_3, \bar{v}_4$ 

が存在する。この参照ブロックに含まれる動きベクトル

【0020】

【外4】

 $\bar{v}_i$  ( $i=1, 2, 3, 4$ )

に対して重み付けし、次式より新たな動きベクトル

【0021】

【外5】

 $\bar{v}_{new}$ 

を算出する。図6に示すように、符号化対象ブロックは  $(n+2)$  フレーム内、予測ブロックは  $(n+1)$  フレーム内に点線で示されたものとして存在する。

【0022】

【数1】

$$\bar{v}_{new} = \sum_{i=1}^4 w_i \cdot \bar{v}_i + \bar{v}_0 \quad \left( \sum_{i=1}^4 w_i = 1 \right)$$

ここで、 $w_i$  は、参照ブロックと予測ブロックとの関係から算出されるものである。例えば、予測ブロックに占める各参照ブロック $block_i$ の面積比率から算出することができる。この他、符号化対象ブロックと各参照ブロックと画像信号の相関係数などがある。

【0023】間引くフレームが複数存在する場合、上記の処理を繰り返すことで新たな動きベクトルを算出することが可能となる。

【0024】上述したように、符号化対象フレームから動きベクトルを辿り新たな参照フレームに対する動きベクトルの算出を行なう。

【0025】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して説明する。

【0026】図1は本発明の一実施形態の、復号信号の符号化方法を示すフローチャートである。

【0027】本実施形態では、間引く参照フレーム数を

4

1フレーム、ブロック形状は正方形とし、新たな動きベクトルを算出するための重み係数は面積比率から算出する。

1. 間引く参照フレームの符号化データから、各参照ブロックの符号化モードと動きベクトルを抽出する（ステップ1）。

2. 符号化対象フレームの符号化データから、各ブロックの符号化モードと動きベクトルを抽出する（ステップ2）。

10 3. ここで、符号化対象ブロックがフレーム内符号化（Intra）で符号化されている場合は、再符号化もIntraモードにて符号化を行なう（ステップ3）。

4. 符号化対象ブロックがフレーム間符号化（Inter）の場合、面積比率から重み係数を計算し、直前に符号化したフレームへの新たな動きベクトルを算出する（ステップ4）。

5. 算出した動きベクトルを用いたInterモードとIntraモードの符号化効率を調べ、符号化モードを決定する（ステップ5）。

20 【0028】図2はステップ4の動きベクトル算出のフローチャートである。

1. 符号化対象ブロックの動きベクトル

【0029】

【外6】

 $\bar{v}_0$ 

から予測ブロックの位置を算出する（ステップ11）。

2. 予測ブロックがまたがる参照フレーム内の1つ以上の参照ブロックを検出する（ステップ12）。予測ブロックの位置により、1～4つの参照ブロック $block_i$  ( $i=1 \sim 4$ ) と重なる可能性がある（図3）。参照ブロック数をNBとする。

30 3. 予測ブロックと各参照ブロック $block_i$  ( $i=1, \dots, NB$ ) が重なりあう面積 $p_i$ を算出する（ステップ13）。

4. 各参照ブロック $block_i$  ( $i=1, \dots, NB$ ) の符号化モードを調べ、Intraモードのブロックを除外する（ステップ14）。

5. Intraモード参照ブロックの占有面積が50%を越える場合、符号化対象ブロックの符号化モードをIntraモードとする（ステップ15）。

6. Interモードの参照ブロックInter- $block_i$  ( $i=1, \dots, NIB < NB$ ) が占める総面積

40 Pallを計算する（ステップ16）。ここで、NIBはInterモード参照ブロックの数、Inter- $p_i$ は各Interモード参照ブロックの占有面積である。

【0030】

【数2】

$$Pall = \sum_{i=1}^{NIB} Inter-p_i$$

5

7. 各Interモード参照ブロックの占有面積Inter-pi、各参照ブロックの動きベクトル

【0031】

【外7】

$$\vec{v}_i$$

、符号化対象ブロックの動きベクトル

【0032】

【外8】

$$\vec{v}_0$$

から、符号化対象ブロックの新たな動きベクトル

【0033】

【外9】

$$\vec{v}_{new}$$

を算出する。

【0034】

【数3】

$$\vec{v}_{new} = \frac{1}{P_{all}} \sum_{i=1}^{NIB} Inter - P_i \cdot \vec{v}_i + \vec{v}_0$$

算出した符号化モードと動きベクトルを用いて符号化処理を行なう。これにより、参照フレームを間引いた場合においても、動き探索を行なうことなく動きベクトルを算出できる。

【0035】以上述べた実施形態では、重み係数として面積比率を用いたが、符号化対象ブロックと各参照ブロックと関係を表すことができる物理量であればよい。また、ブロックを正方形としたが、これに限定されるものではない。

【0036】ここで、符号化対象ブロックの新たな動きベクトル

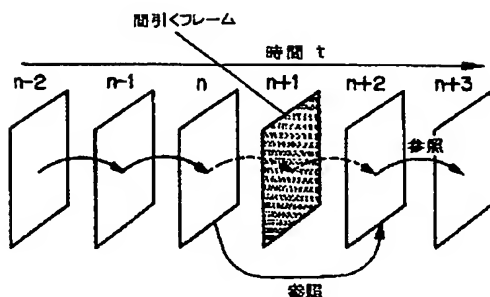
【0037】

【外10】

$$\vec{v}_{new}$$

の算出例を図4により説明する。n+1フレームにおいて符号化対象ブロックが参照するブロックを含む4ブロックを決める。n+1フレームとnフレームの比較によ

【図5】



6

って得られている、これら4ブロックblock1、block2、block3、block4が有する動きベクトル

【0038】

【外11】

$$\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4$$

をn+1フレームの符号化データから得る。block1、block2、block3、block4の面積比を計算する。これらの面積比を $w_1 = 0.1$ ,  $w_2 = 0.6$ ,  $w_3 = 0.05$ ,  $w_4 = 0.25$ とすると、新たな動きベクトル

【0039】

【数4】

$$\vec{v}_{new} = \vec{v}_0 + 0.1\vec{v}_1 + 0.6\vec{v}_2 + 0.05\vec{v}_3 + 0.25\vec{v}_4$$

が得られる。

【0040】

【発明の効果】以上説明したように、本発明によれば、再符号化時に、動き検出をすることなく参照フレームを間引くことができ、結果として演算量を増加させず、かつ、画質劣化を抑えた符号化データのスケールリングが可能となる。

【図面の簡単な説明】

【図1】本発明の一実施形態の、復号信号の符号化方法のフローチャートである。

【図2】図1中の動きベクトル算出のフローチャートである。

【図3】予測ブロックと重なる参照ブロックを示す図である。

【図4】動きベクトル算出の具体例を示す図である。

【図5】フレーム間の参照関係を示す図である。

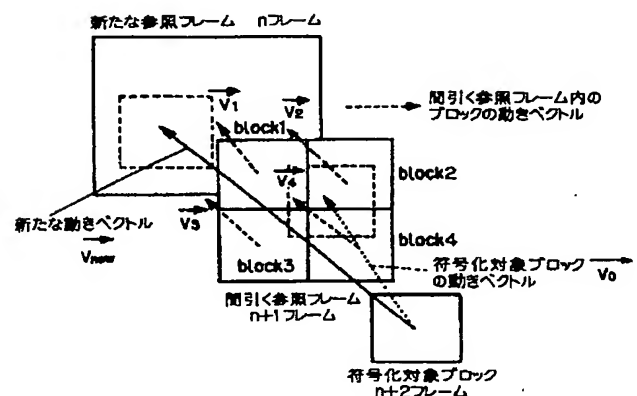
【図6】符号化対象ブロックと参照フレームの関係を示す図である。

【図7】符号化データの再符号化を示す図である。

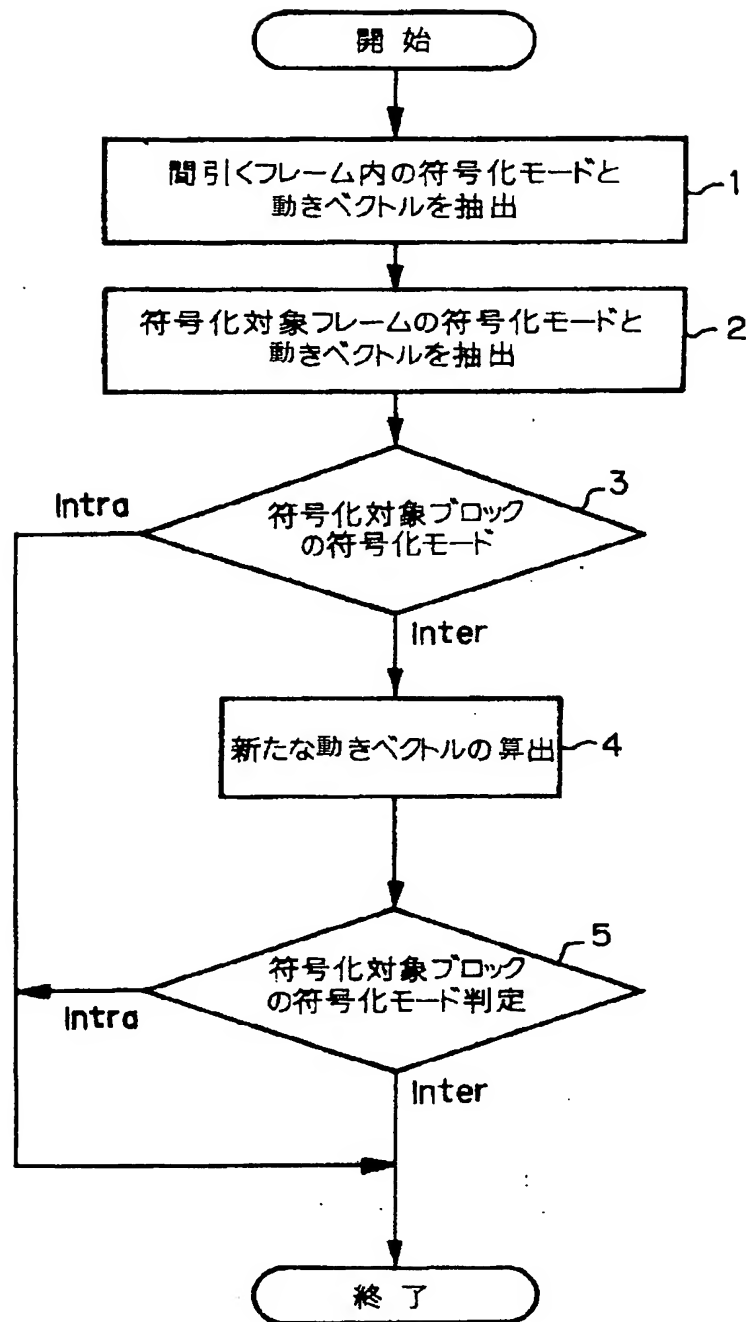
【符号の説明】

1～5, 11～17 ステップ

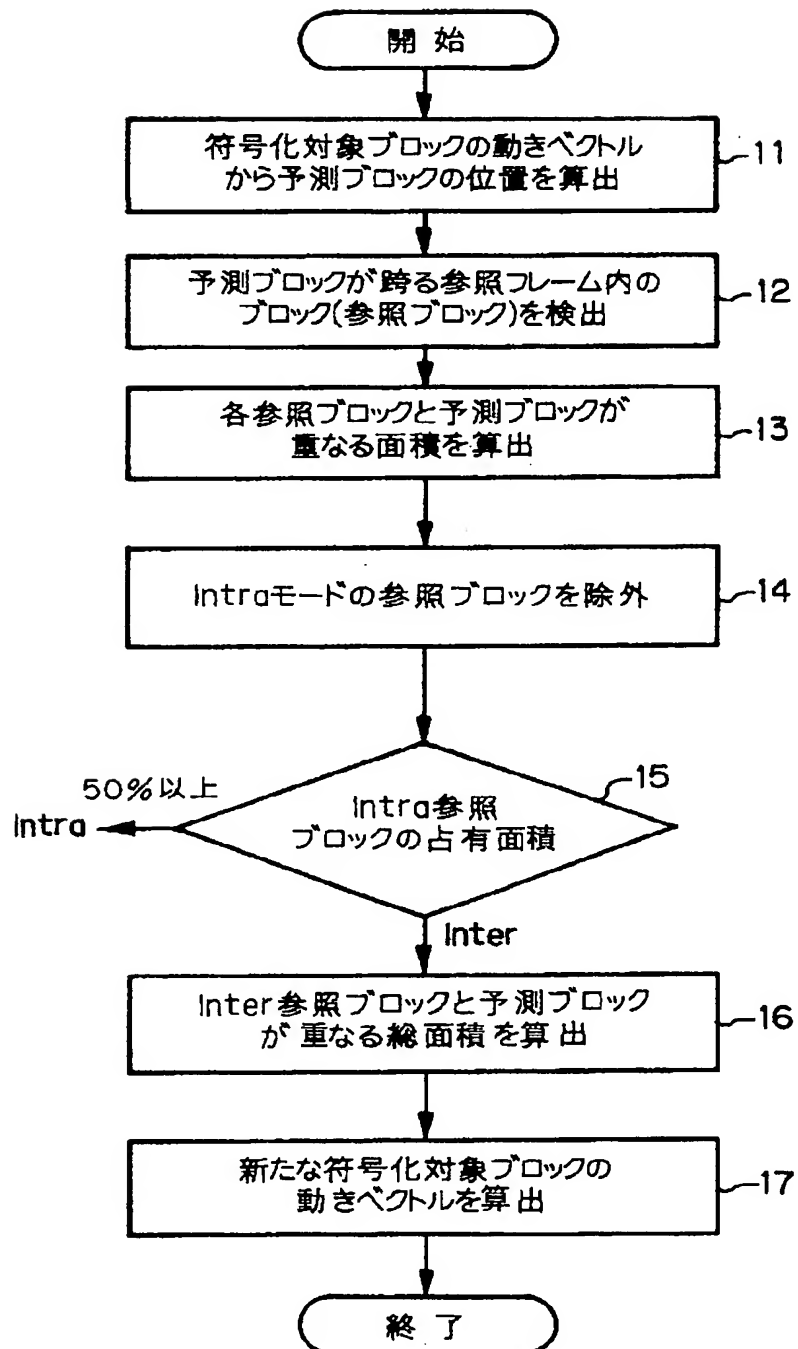
【図6】



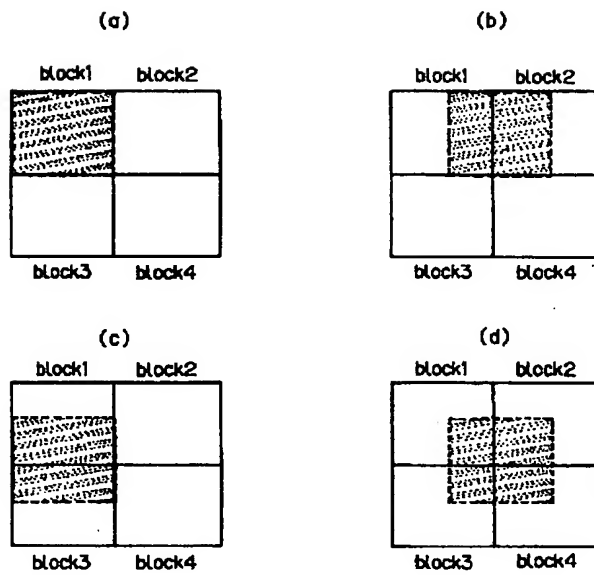
【図1】



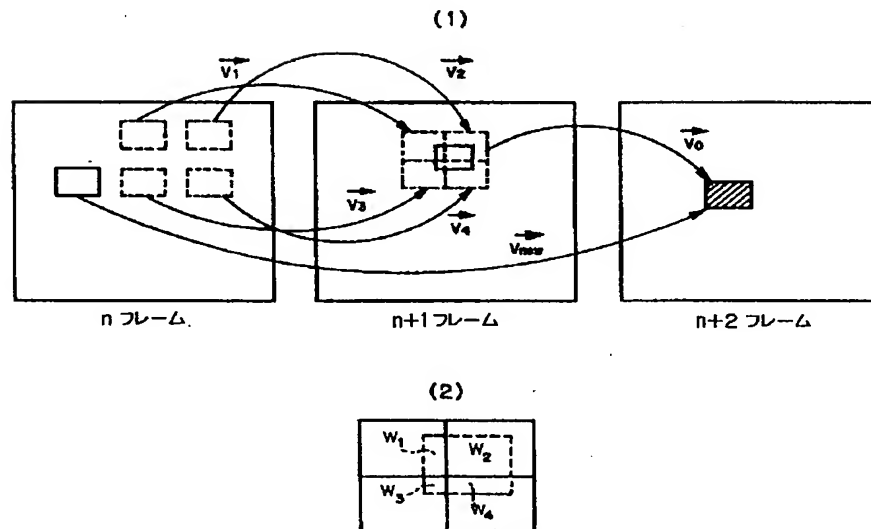
【図2】



【図3】



【図4】



【図7】

